Guide for students

to be read after the Guide for new users

(last updated 9. 10.2019, Lenka Forstová)

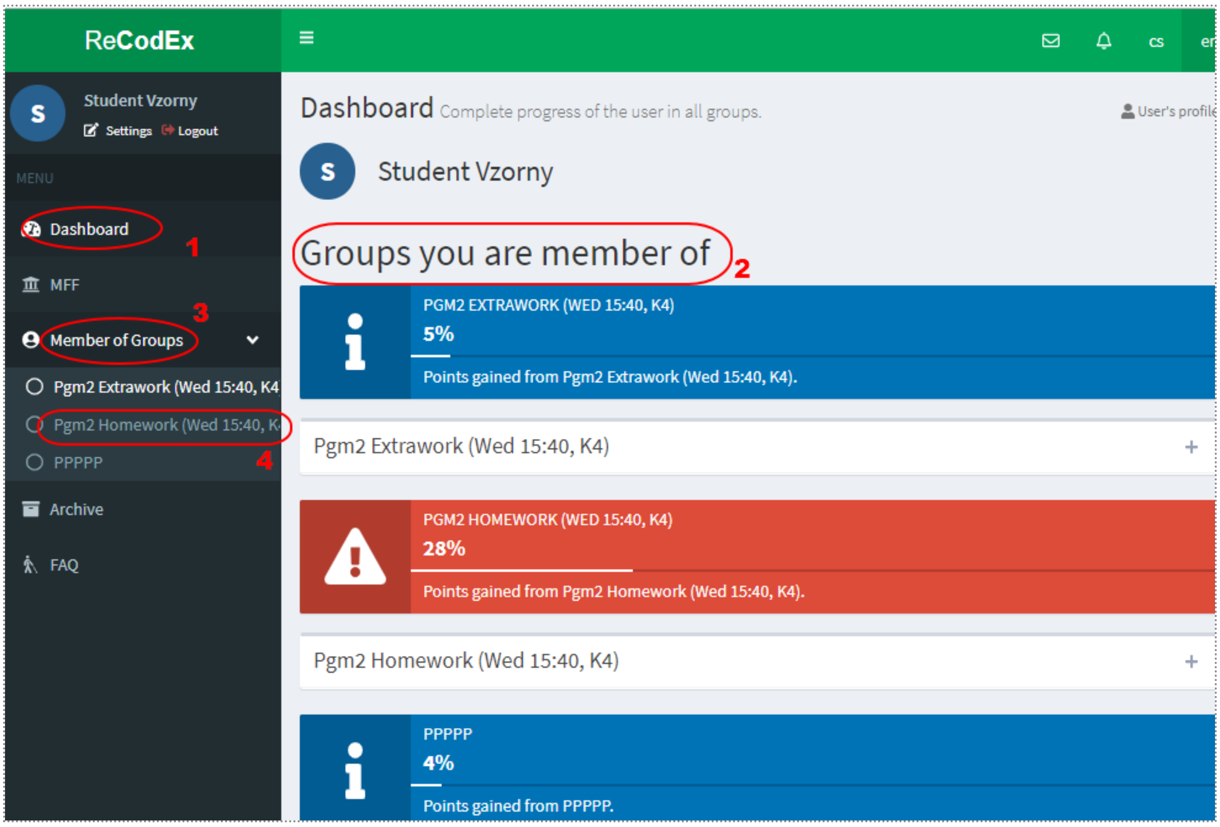
Reporting bugs and problems

If you run into problems while working with ReCodEx and neither the documentation nor your supervisor can help, you may contact us at recodex@mff.cuni.cz

Dashboard

The *Dashboard* page **(1.1)** is a basic page that appears after you log into ReCodEx, unless you have specified otherwise on the *Settings* page.

This page will change substantially in the near future, and so for now we will not describe it in detail.



Dashboard Complete progress of the user in all groups.

Student Vzorny

Groups you are member of

PGM2 EXTRAWORK (WED 15:40, K4)
5%
Points gained from Pgm2 Extrawork (Wed 15:40, K4).

Pgm2 Extrawork (Wed 15:40, K4)

PGM2 HOMEWORK (WED 15:40, K4)
28%
Points gained from Pgm2 Homework (Wed 15:40, K4).

Pgm2 Homework (Wed 15:40, K4)

PPPPP
4%
Points gained from PPPPP.

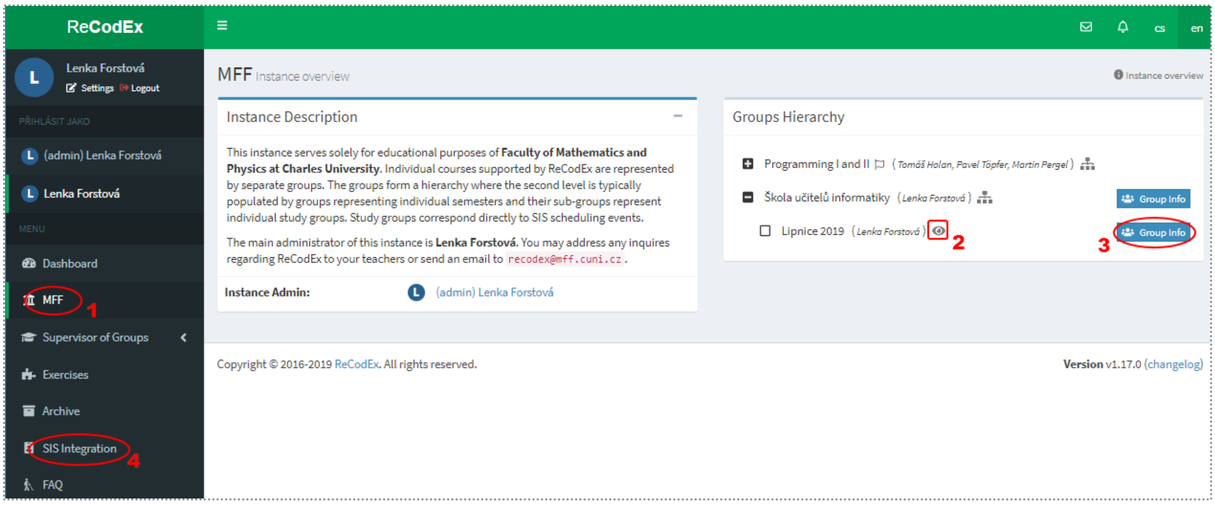
Figure 1

Joining groups

The list of groups that you belong to **(1.2)** appears on the *Dashboard* page **(1.1)** or in the sidebar on the left **(1.3)**.

- A group's supervisor may add you directly to the group. In that case there is nothing for you to do.

- You can join a group that belongs to a class that you have registered for in SIS. In the sidebar on the left, choose the item *SIS integration* **(2.4)** and choose a group corresponding to one of your scheduled classes.



MFF Instance overview

Instance Description

This instance serves solely for educational purposes of **Faculty of Mathematics and Physics at Charles University**. Individual courses supported by ReCodEx are represented by separate groups. The groups form a hierarchy where the second level is typically populated by groups representing individual semesters and their sub-groups represent individual study groups. Study groups correspond directly to SIS scheduling events.

The main administrator of this instance is **Lenka Forstová**. You may address any inquires regarding ReCodEx to your teachers or send an email to `recodex@mff.cuni.cz`.

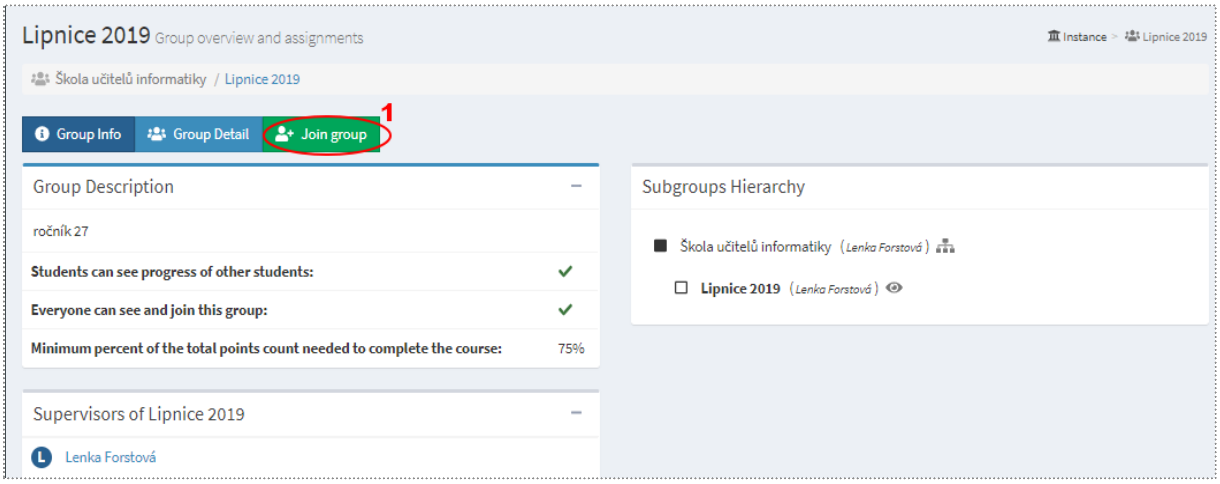
Instance Admin: (admin) Lenka Forstová

Copyright © 2016-2019 ReCodEx. All rights reserved.

Version v1.17.0 (changelog)

Figure 2

- Or you can join a public group – a group whose supervisor has marked it as public, allowing anyone with a ReCodEx account to join it. In this case
    1. on the instance page ("MFF" in our case) **(2.1)**
    2. choose a group that is marked as public **(2.2)**
    3. display information about this group **(2.3)**
    4. and join the group on its information page **(3.1)**



Lipnice 2019 Group overview and assignments

Škola učitelů informatiky / Lipnice 2019

Group Description

ročník 27

Students can see progress of other students:	✓
Everyone can see and join this group:	✓
Minimum percent of the total points count needed to complete the course:	75%

Supervisors of Lipnice 2019

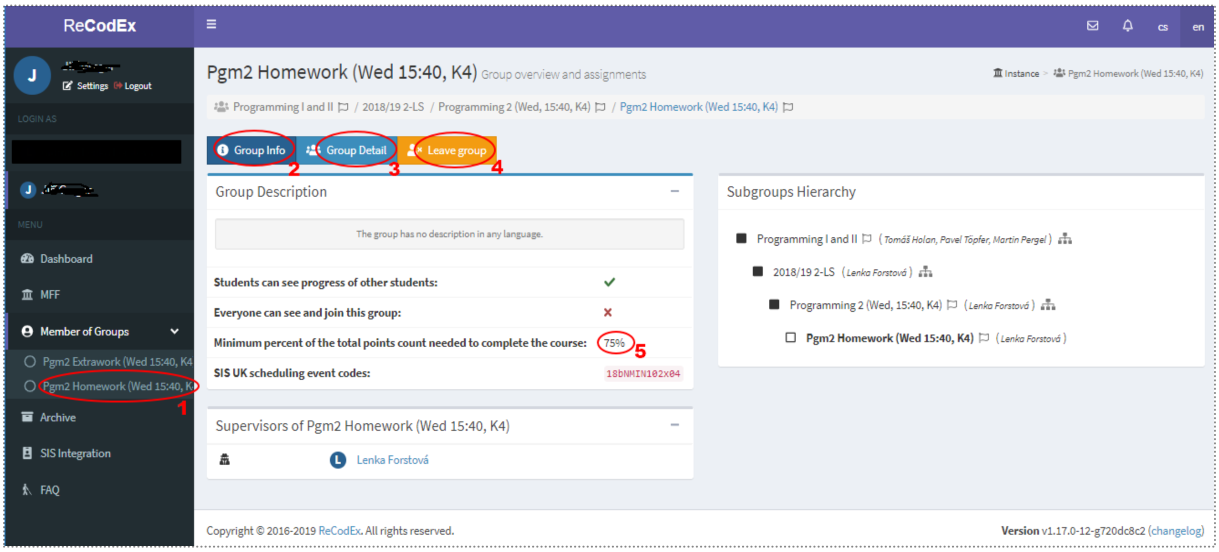
Lenka Forstová

Figure 3

Membership in a group

Choose a specific group, e.g. in the sidebar on the left **(4.1)** to display concrete information about the group, divided into two pages *Group Info* **(4.2)** and *Group Detail* **(4.3)**. The third tab *Leave Group* **(4.4)** is a command to terminate your membership in a group.

On the *Group Info* page you will find the name of the group's supervisor and the group's attributes set by the supervisor. For students, the most important of these atttributes determines the minimum number of points needed to pass **(4.5)**. This number is determined by a percentage of points that students must earn to fulfill the supervisor's requirements. This percentage is based on the total number of points that a student can earn by completing assignments by their first assigned deadline. Bonus assignments do not count toward this base total. The base total continually increases as the supervisor assigns more exercises. As a student gains points by solving exercises, ReCodex continually updates the student's success percentage and the *Dashboard* uses colors to show whether the student is above or below the required minimal threshold. A group's supervisor may, but is not required to, set this „minimum percent“ attribute.



Pgm2 Homework (Wed 15:40, K4) Group overview and assignments

Programming I and II / 2018/19 2-LS / Programming 2 (Wed, 15:40, K4) / Pgm2 Homework (Wed 15:40, K4)

Group Description

The group has no description in any language.

Students can see progress of other students:	✓
Everyone can see and join this group:	✗
Minimum percent of the total points count needed to complete the course:	75%
SIS UK scheduling event codes:	18bNMIN102x04

Supervisors of Pgm2 Homework (Wed 15:40, K4)

L Lenka Forstová

Subgroups Hierarchy

- ■ Programming I and II (*Tomáš Holan, Pavel Töpfer, Martin Pergel*)
  - ■ 2018/19 2-LS (*Lenka Forstová*)
    - ■ Programming 2 (Wed, 15:40, K4) (*Lenka Forstová*)
      - □ **Pgm2 Homework (Wed 15:40, K4)** (*Lenka Forstová*)

Copyright © 2016-2019 ReCodEx. All rights reserved.

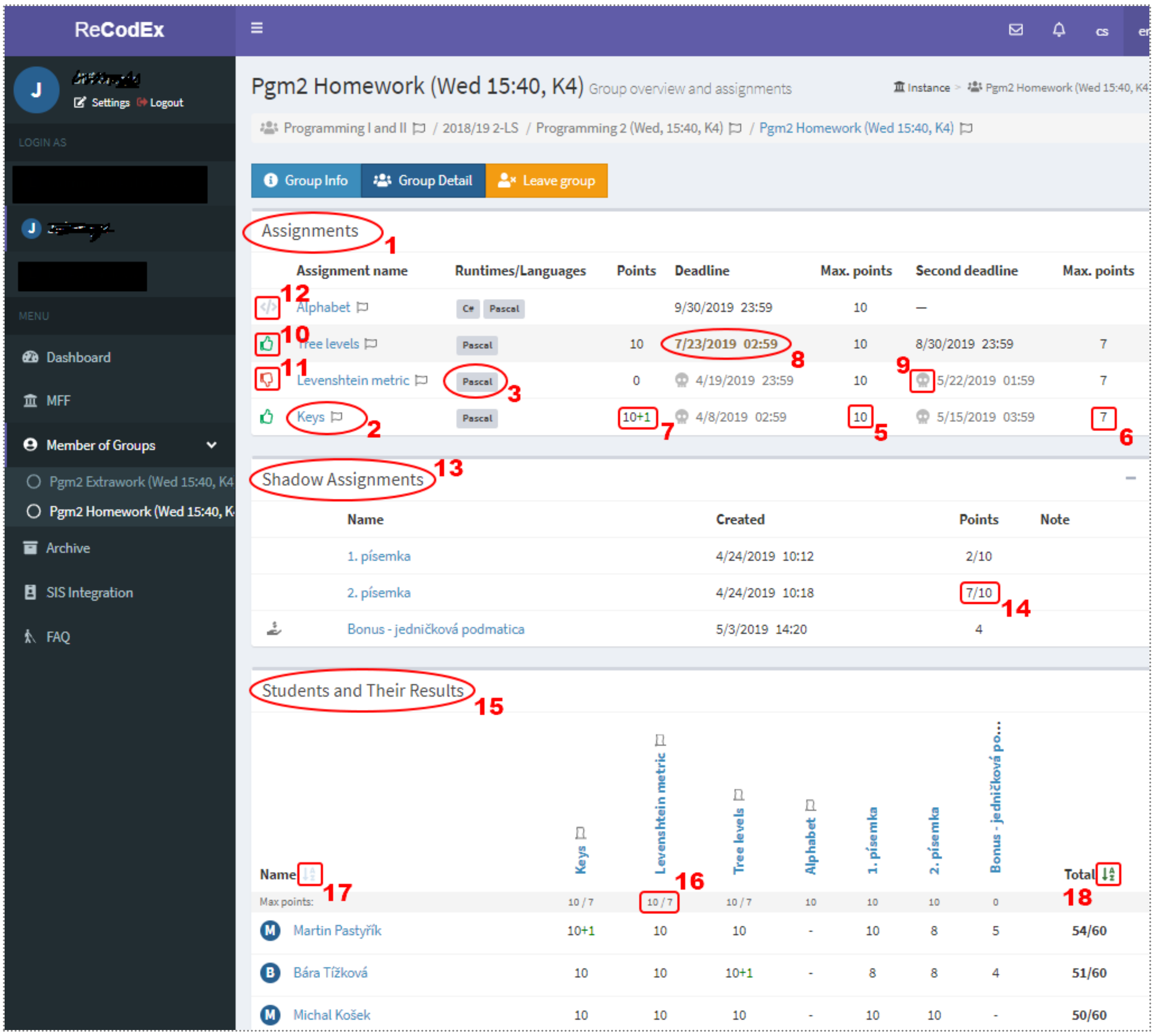
Version v1.17.0-12-g720dc8c2 (changelog)

Figure 4

Overview of assignments

A group's supervisor assigns exercises to its members. Members can see an overview of assignments on the Group Detail page (see above) in the *Assignments* table **(5.1)**. Each row contains:

- The assignment's name **(5.2)** – you can click this to see details about the assignment (which will be described later)
- Runtimes/languages **(5.3)** – the language(s) in which you may submit a solution to the assignment



Pgm2 Homework (Wed 15:40, K4) Group overview and assignments

Programming I and II / 2018/19 2-LS / Programming 2 (Wed, 15:40, K4) / Pgm2 Homework (Wed 15:40, K4)

Assignments

	Assignment name	Runtimes/Languages	Points	Deadline	Max. points	Second deadline	Max. points
</>	Alphabet	C# Pascal		9/30/2019 23:59	10	—	
✓	Tree levels	Pascal	10	7/23/2019 02:59	10	8/30/2019 23:59	7
✗	Levenshtein metric	Pascal	0	4/19/2019 23:59	10	5/22/2019 01:59	7
✓	Keys	Pascal	10+1	4/8/2019 02:59	10	5/15/2019 03:59	7

Shadow Assignments

	Name	Created	Points	Note
	1. písemka	4/24/2019 10:12	2/10	
	2. písemka	4/24/2019 10:18	7/10	
	Bonus - jedničková podmatica	5/3/2019 14:20	4	

Students and Their Results

Name	Keys	Levenshtein metric	Tree levels	Alphabet	1. písemka	2. písemka	Bonus - jedničková po...	Total
Max points:	10 / 7	10 / 7	10 / 7	10	10	10	0	
M Martin Pastyřík	10+1	10	10	-	10	8	5	54/60
B Bára Tížková	10	10	10+1	-	8	8	4	51/60
M Michal Košek	10	10	10	-	10	10	-	50/60

Figure 5 (21. 7. 2019 12:00)

Points – the number of points that a student can earn by the first **(5.5)** or second **(5.6)** deadline, and the number of points that a student has already earned for this assignment **(5.7)**. A supervisor may award bonus points (positive or negative) for an submitted solution. Earned points are displayed as points automatically awarded by ReCodEx plus bonus points awarded by the supervisor **(5.7)**. The sum of automatically awarded points and bonus points is included in a student's point total.
Deadlines that have passed **(5.9)** and upcoming deadlines **(5.8)** are visually distinguished. A student may submit solutions even after the last deadline has passed, but in this case 0 points will be awarded for the assignment.

A solution may also receive zero points if it does not achieve the minimum success percentage assigned by the supervisor in the output tests.

The page visually indicates whether a student has

- successfully solved the assignment, i.e. achieved the minimal required percentage in the tests **(5.10)**
- submitted a solution, but unsuccessfully **(5.11)**
- not submitted any solution **(5.12)**

The second part of the *Group Detail* displays a list of results of *Shadow assignments* **(5.12)**. These are assignments that are not graded by ReCodEx; the supervisor grades them, but points earned for them are counted normally in a student's point total for the group. Points received **(5.14)** are displayed in the form of points awarded out of the maximum possible points (if it is set).

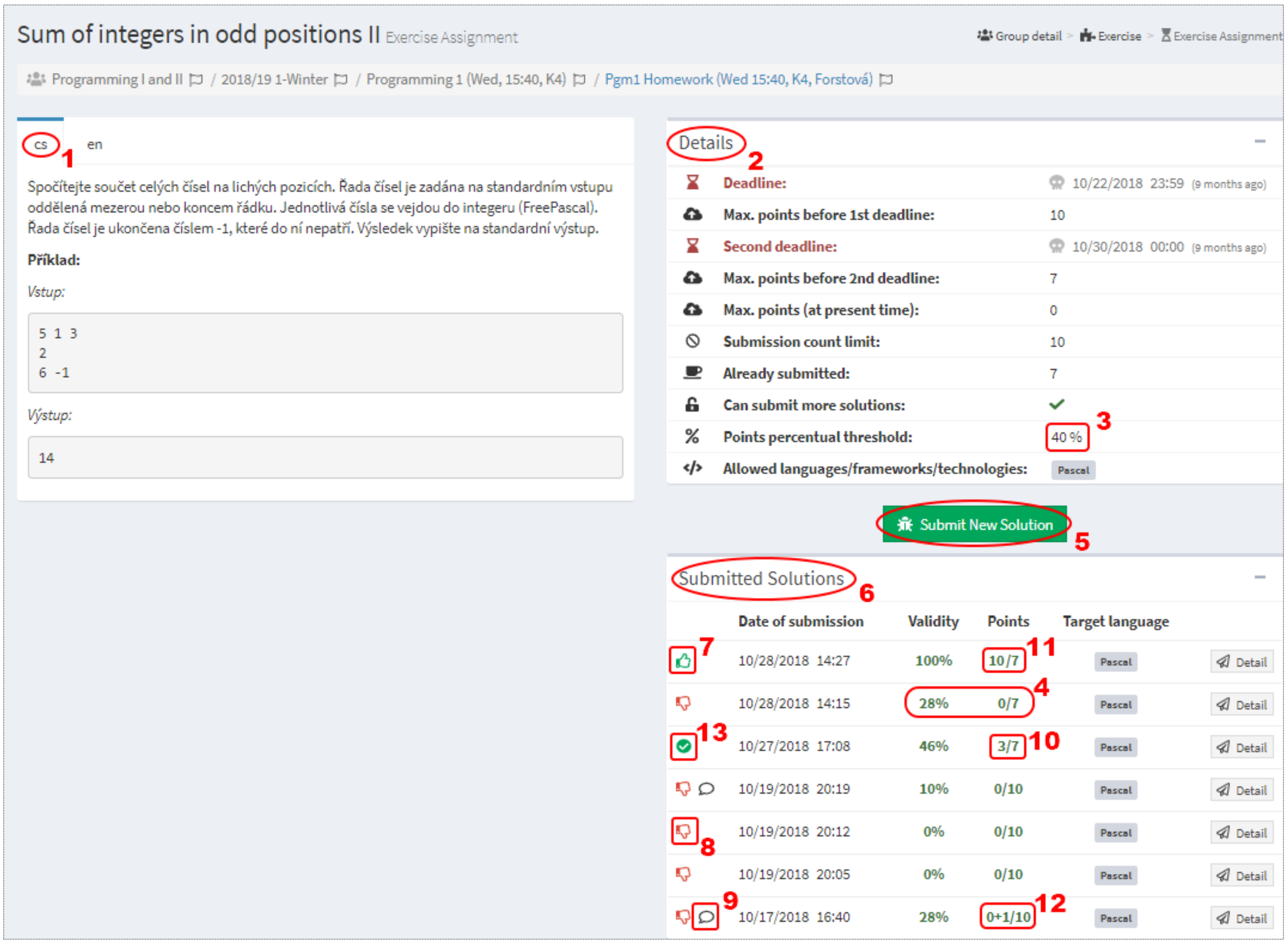
In the third section are *Students and their results* **(5.15)**. The table heading displays each assignment along with its maximum number of points that can be earned by the first / second deadline **(5.16)**

A group's supervisor may decide whether students can see each other's results. If this has been enabled, each member of the group sees all other members' results here.

You can sort the table (in ascending or descending order) by the last name of group members **(5.17)** or by the number of points achieved **(5.18)**.

Assignment details

Click on the name of any assignment to display a page with details about it. You can display text about the assignment in any available language **(6.1)**.



Sum of integers in odd positions II Exercise Assignment

Group detail > Exercise > Exercise Assignment

Programming I and II / 2018/19 1-Winter / Programming 1 (Wed, 15:40, K4) / Pgm1 Homework (Wed 15:40, K4, Forstová)

cs en

Spočítejte součet celých čísel na lichých pozicích. Řada čísel je zadána na standardním vstupu oddělená mezerou nebo koncem řádku. Jednotlivá čísla se vejdou do integeru (FreePascal). Řada čísel je ukončena číslem -1, které do ní nepatří. Výsledek vypište na standardní výstup.

Příklad:

Vstup:

```
5 1 3
2
6 -1
```

Výstup:

```
14
```

Details

⧗	**Deadline:**	10/22/2018 23:59 (9 months ago)
	Max. points before 1st deadline:	10
⧗	**Second deadline:**	10/30/2018 00:00 (9 months ago)
	Max. points before 2nd deadline:	7
	Max. points (at present time):	0
⃠	**Submission count limit:**	10
	Already submitted:	7
	Can submit more solutions:	✓
%	**Points percentual threshold:**	40 %
</>	**Allowed languages/frameworks/technologies:**	Pascal

Submit New Solution

Submitted Solutions

	Date of submission	Validity	Points	Target language	
👍	10/28/2018 14:27	100%	10/7	Pascal	Detail
👎	10/28/2018 14:15	28%	0/7	Pascal	Detail
✓	10/27/2018 17:08	46%	3/7	Pascal	Detail
👎 💬	10/19/2018 20:19	10%	0/10	Pascal	Detail
👎	10/19/2018 20:12	0%	0/10	Pascal	Detail
👎	10/19/2018 20:05	0%	0/10	Pascal	Detail
👎 💬	10/17/2018 16:40	28%	0+1/10	Pascal	Detail

Figure 6

Read the assignment carefully; many errors arise by not following an assignment or by misinterpreting it (a common mistake is that a student assumes a particular input format, but actually the assignment is more general). If an assignment includes a filename, you must use precisely the same name in your solution, including its extension and case.

Details table

The *Details* table displays attributes about an assignment and your solution to it **(6.2)**.

- Deadlines set by the supervisor, along with the remaining time until (or time elapsed since) each deadline and the number of points that can be earned by it.
- The maximum number of points that can be earned by submitting a solution at the present time. Beware: this is not the maximum number of points that it was possible to earn by submitting by an appropriate deadline, which is the value that contributes toward a student's success percentage.
- The submission count limit, set by the supervisor. If a student reaches this limit, they may not submit further solutions. A supervisor may delete selected submissions by a student and thus enable further submissions..
- The number of solutions already submitted
- An indicator of whether it is still possible to submit solutions. A student can submit solutions even after the deadline has passed, as long as the supervisor has not disallowed it and the student has not reached the submission limit.
- The points percentual threshold, set by the supervisor **(6.3)**. If a solution does not achieve at least this threshold, it receives 0 points **(6.4)**.
- Allowed environments/languages. The supervisor sets the language(s) in which solutions may be submitted.

Submitted solutions

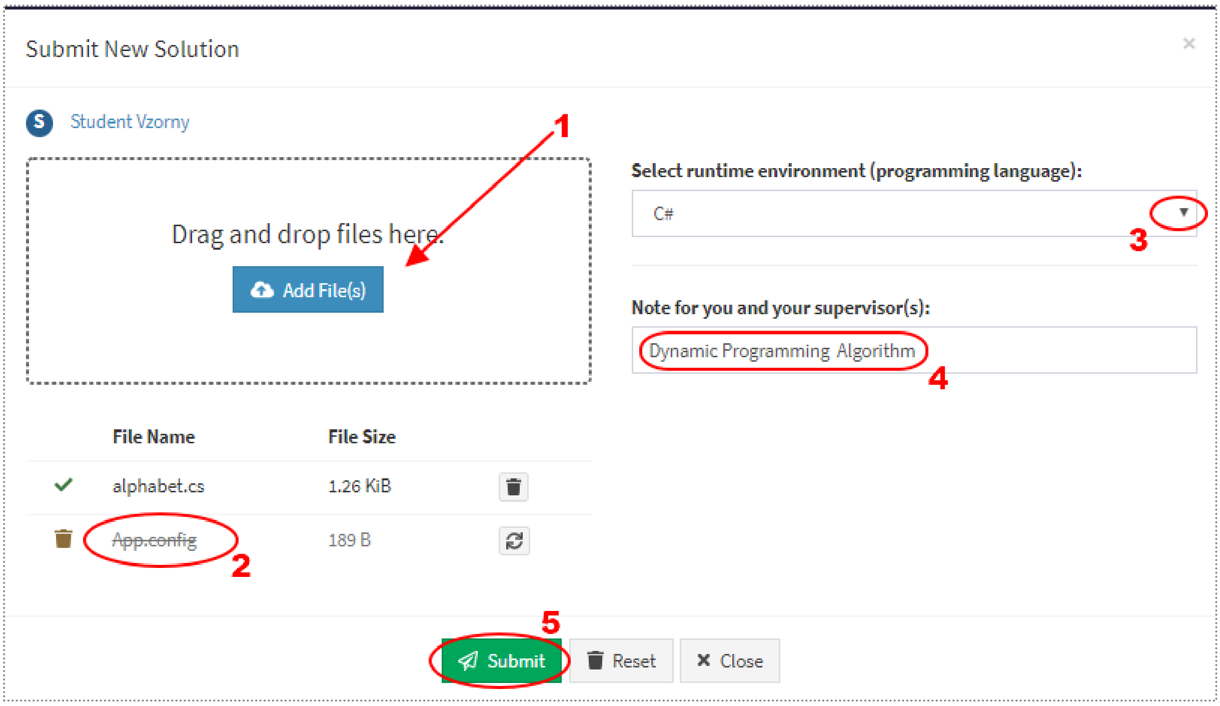
To submit a solution, click the *Submit New Solution* button **(6.5**).

You will find an overview of submitted solutions in the *Odevzdaná řešení* table **(6.6)**. Icons in the first table column indicate whether a solution has met **(6.7)** or has not met **(6.8)** the required point threshold **(6.3)** and whether a submitted solution was annotated with comments **(6.9)**. Either the supervisor or student may add comments to a solution at any time. The points column displays the number of points awarded out of the total points **(6.10)** that it was possible to earn when the assignment was submitted. A supervisor may assign a point total outside the set range **(6.11)**, e.g. when an assignment was submitted late for excusable reasons, or a supervisor may award bonus points for a successful solution **(6.12)**.
A supervisor may denote a submitted solution as *final* **(6.13**). In this case, the points from this point are counted in the student's point total even if the student submits another solution (possibly in the future) that gains more points. In the example in the screenshot above, the student earns 3 points for the assignment. A supervisor may remove the *final* designation at any point, and then the student will once again receive the maximum number of points from any submitted solution, in this case 10 points **(6.11)**.

Click the *Detail* button **(6.11)** to display the *Solution evaluation* page **(Figure 8)**, where you can peruse e.g. the compilation logs or the results of individual tests.

Submitting a solution

When you click the *Submit New Solution* button **(6.5)** on the assignment details page, a dialog appears that is used to submit your solution to the system. You can drag file(s) into the indicated area or click the *Add File(s)* button **(7.1)** to launch a file browser dialog. The name(s) of the file(s) should not contain non-standard characters. You will definitely be safe if you stick to the letters of the English alphabet, digits and the underscore character. A file should have an extension corresponding to the chosen runtime environment. For Pascal, for example, this is the extension *pas* or *lpr*. Only submit files that ReCodEx needs for testing your solution; do not submit e.g. files for configuring your IDE **(7.2)**. ReCodEx will attempt to test these files and will needlessly crash on inappropriate files. Each crash launches an error process, generate an error log, and is an unnecessary burden on the system and its administrators.



Submit New Solution

Student Vzorny

Select runtime environment (programming language): C#

Note for you and your supervisor(s): Dynamic Programming Algorithm

	File Name	File Size
✓	alphabet.cs	1.26 KiB
	~~App.config~~	189 B

Figure 7

After you submit file(s) with your solution, ReCodEx will attempt to determine the language (runtime environmennt) in which your files should be tested. You may modify this choice **(7.3)**.

You may attach a note to your solution **(7.4)**, which will appear in the table of submitted solutions and may help you distinguish between individual variants of your solution. The group supervisor will also see thsi note. It it not possible to modify or edit this note later.

To initiate the evaluation process, click the *Submit* button **(7.5)**.

Instructions and advice:

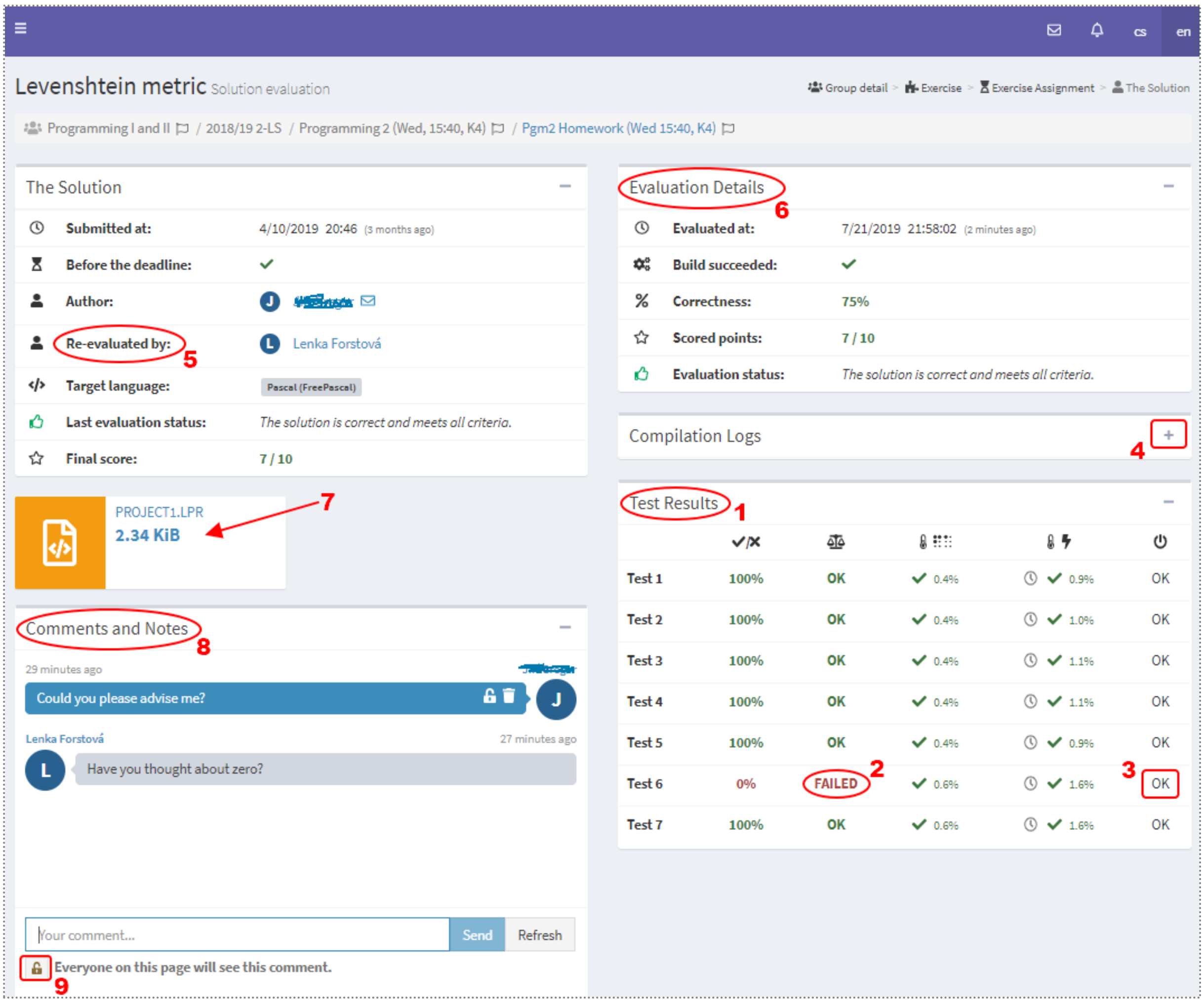
- *ReCodEx is a system intended for checking assignment solutions, not a system intended for debugging programs. You should only submit solutions that you have already thoroughly debugged and have tested on various inputs.*
- *Do not use characters from non-English alphabets in source files.*
- *In performing your own testing, it is best to use the same compiler that ReCodEx uses. You can see the name and version number of this compiler at the beginning of the compilation log on the* Solution evaluation *page, or look at the project wiki linked from the FAQ page at https://github.com/ReCodEx/wiki/wiki/FAQ.*

Solution evaluation

As soon as you submit a solution **(7.5)** the system will evaluate it. Each exercise's author has prepared a suite of tests. Your solution is compiled and is then launched separately on each test. For each test, ReCodEx determines whether the submitted solution ran in the predetermined conditions with the expected result, using less time and memory than the limits set by the author. The results of all tests are displayed in an overview table **(8.1)** on the *Solution evaluation* page.

A student invokes the process of testing their solution only by submitting it **(7.5)**. The gropu supervisor may explicitly launch a solution evaluation whenever they deem it appropriate **(8.5)**. For example, a supervisor might do this when tests have been added or amended, or when they need additional debugging information (an expanded log file) for a given evaluation. Nevertheless the *Evaluation Details* **(8.6)** and the *Test Results* table **(8.1)**, which are displayed to the student, always reflect the evaluation that ran last, whether it was initiated by the student or the supervisor.

The Solution evaluation page also contains a link to a file with source code **(8.7)** and the frame *Comments and Notes* **(8.8)**, which is intended primarily for communication between a student and their advisor regarding a concrete solution submssion. If you mark the comments as private **(8.9)**, only you will see them.



Levenshtein metric Solution evaluation

Group detail > Exercise > Exercise Assignment > The Solution

Programming I and II / 2018/19 2-LS / Programming 2 (Wed, 15:40, K4) / Pgm2 Homework (Wed 15:40, K4)

The Solution

Submitted at:	4/10/2019 20:46 (3 months ago)
Before the deadline:	✓
Author:	J [redacted]
Re-evaluated by:	L Lenka Forstová
Target language:	Pascal (FreePascal)
Last evaluation status:	The solution is correct and meets all criteria.
Final score:	7 / 10

PROJECT1.LPR
2.34 KiB

Comments and Notes

29 minutes ago [redacted]
Could you please advise me?

Lenka Forstová 27 minutes ago
Have you thought about zero?

Evaluation Details

Evaluated at:	7/21/2019 21:58:02 (2 minutes ago)
Build succeeded:	✓
Correctness:	75%
Scored points:	7 / 10
Evaluation status:	The solution is correct and meets all criteria.

Compilation Logs

Test Results

	✓/✗	⚖	🌡	🌡 ⚡	⏻
Test 1	100%	OK	✓ 0.4%	✓ 0.9%	OK
Test 2	100%	OK	✓ 0.4%	✓ 1.0%	OK
Test 3	100%	OK	✓ 0.4%	✓ 1.1%	OK
Test 4	100%	OK	✓ 0.4%	✓ 1.1%	OK
Test 5	100%	OK	✓ 0.4%	✓ 0.9%	OK
Test 6	0%	FAILED	✓ 0.6%	✓ 1.6%	OK
Test 7	100%	OK	✓ 0.6%	✓ 1.6%	OK

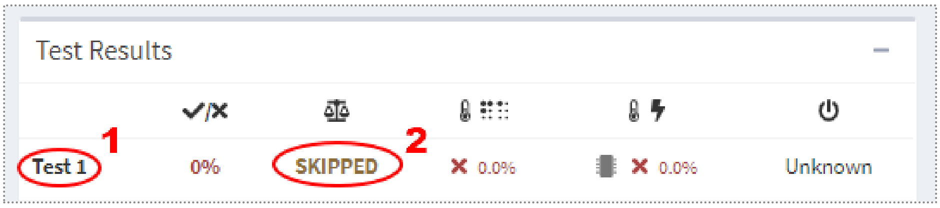
Everyone on this page will see this comment.

Figure 8

Test results

The meaning of individual table columns **(8.1)**:

- The name of the test **(9.1),** according to which a supervisor can set the test configuration. So if you wish to discuss a particular test with your supervisor, you should refer to the test by this name. The supervisor can verify the correctness of the test even if they are not the author of the exercise.

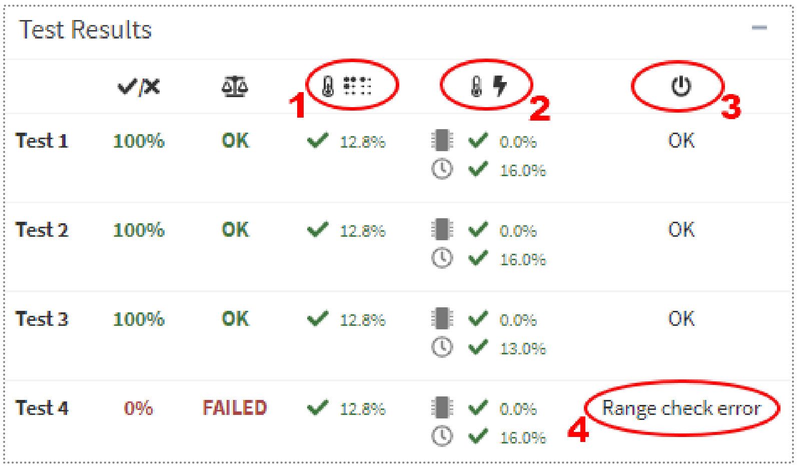


Test Results

	✓/✗	⚖	🌡 ⠿	🌡 ⚡	⏻
Test 1	0%	SKIPPED	✗ 0.0%	▮ ✗ 0.0%	Unknown

Figure 9

- The percentage of correctness of any individual test in most cases is either 0 or 100.
- The result of comparing the submitted solution's output with the author's solution:
    - *Skipped* **(9.2)** indicates that the program did not run, e.g. due to a syntax error or because the compiler of the chosen language failed. In this case we recommend studying the *Compilation logs* **(8.4)**
    - *Failed* **(8.2)** indicates that the submitted solution ran successfully (there was no runtime error) **(8.3)**, but did not achieve the expected result, i.e. for the given input the program being tested produced output that differed from the expected output provided by the exercise author
- The percentage of the memory limit that was used **(10.1)** (the supervisor may hide this data from students)



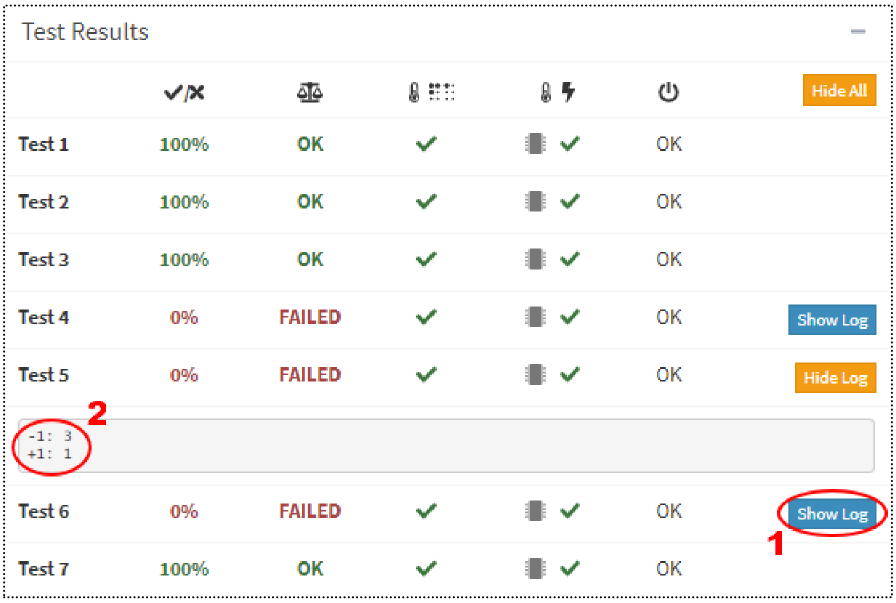
Test Results

	✓/✗	⚖	🌡 ⠿	🌡 ⚡	⏻
Test 1	100%	OK	✓ 12.8%	▮ ✓ 0.0% 🕓 ✓ 16.0%	OK
Test 2	100%	OK	✓ 12.8%	▮ ✓ 0.0% 🕓 ✓ 16.0%	OK
Test 3	100%	OK	✓ 12.8%	▮ ✓ 0.0% 🕓 ✓ 13.0%	OK
Test 4	0%	FAILED	✓ 12.8%	▮ ✓ 0.0% 🕓 ✓ 16.0%	Range check error

Figure 10

- The percentage of the time limit that was used **(10.2)** (the supervisor may hide this data from students)
- The result of running the program **(10.3)**, or a description of a runtime error if one occurred **(10.4)**

- For a given exercise, the supervisor may allow students to view the *assignment evaluation log* **(11.1)**, which may help students find bugs in cases where their program runs without error but does not produce the expected result. The *Log* is a text file containing the differences between the expected output and the actual output of the program being tested.



Test Results

	✓/✗	⚖			⏻	Hide All
Test 1	100%	OK	✓	✓	OK	
Test 2	100%	OK	✓	✓	OK	
Test 3	100%	OK	✓	✓	OK	
Test 4	0%	FAILED	✓	✓	OK	Show Log
Test 5	0%	FAILED	✓	✓	OK	Hide Log
Test 6	0%	FAILED	✓	✓	OK	Show Log
Test 7	100%	OK	✓	✓	OK	

```
-1: 3
+1: 1
```

Figure 11

- In the log **(11.2),** the number before the colon indicates a line number in the program's output file; following the colon is the text string belonging to the given line. If a line number is negative, that means that the given value was expected on that line, but it was not there. A positive line number indicates a value that actually appeared on the given line. In the given example, the log indicates that the number 3 should have been in the output, but actually the number 1 appeared there instead.
- Lines on which the output agreed with the expected output do not appear in the log at all.
- If there are too many differences, the log will show only some of them.